<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>高豪商城平台</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>高豪商城平台</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item active"><span>商品浏览</span></li></a> <a ><li class="rz-nav-item"><span>购物车管理</span></li></a> <a ><li class="rz-nav-item"><span>订单处理</span></li></a> <a ><li class="rz-nav-item"><span>支付功能</span></li></a> <a ><li class="rz-nav-item"><span>客户评价</span></li></a> <a ><li class="rz-nav-item"><span>促销活动</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>商品浏览</h2> <div class="filter"> <input type="text" placeholder="搜索商品..." /> <button>搜索</button> </div> <div class="product"> <img src="https://via.placeholder.com/100" alt="商品图片"> <div class="product-details"> <div class="product-title">商品名称 1</div> <div>描述: 这是商品1的详细描述。</div> <div>价格: ¥99.00</div> <a >查看详情</a> </div> </div> <div class="product"> <img src="https://via.placeholder.com/100" alt="商品图片"> <div class="product-details"> <div class="product-title">商品名称 2</div> <div>描述: 这是商品2的详细描述。</div> <div>价格: ¥199.00</div> <a >查看详情</a> </div> </div> <div class="product"> <img src="https://via.placeholder.com/100" alt="商品图片"> <div class="product-details"> <div class="product-title">商品名称 3</div> <div>描述: 这是商品3的详细描述。</div> <div>价格: ¥299.00</div> <a >查看详情</a> </div> </div> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>高豪商城平台</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>高豪商城平台</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>商品浏览</span></li></a> <a ><li class="rz-nav-item active"><span>购物车管理</span></li></a> <a ><li class="rz-nav-item"><span>订单处理</span></li></a> <a ><li class="rz-nav-item"><span>支付功能</span></li></a> <a ><li class="rz-nav-item"><span>客户评价</span></li></a> <a ><li class="rz-nav-item"><span>促销活动</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>购物车管理</h2> <table> <thead> <tr> <th>商品名称</th> <th>单价</th> <th>数量</th> <th>小计</th> <th>操作</th> </tr> </thead> <tbody> <tr> <td>商品A</td> <td>¥100</td> <td> <input type="number" value="1" min="1" style="width: 50px;"> </td> <td>¥100</td> <td class="action-buttons"> <button>删除</button> </td> </tr> <tr> <td>商品B</td> <td>¥200</td> <td> <input type="number" value="1" min="1" style="width: 50px;"> </td> <td>¥200</td> <td class="action-buttons"> <button>删除</button> </td> </tr> </tbody> </table> <div class="total">总计: ¥300</div> <div class="action-buttons"> <button>继续购物</button> <button>结算</button> </div> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>高豪商城平台</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>高豪商城平台</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>商品浏览</span></li></a> <a ><li class="rz-nav-item"><span>购物车管理</span></li></a> <a ><li class="rz-nav-item active"><span>订单处理</span></li></a> <a ><li class="rz-nav-item"><span>支付功能</span></li></a> <a ><li class="rz-nav-item"><span>客户评价</span></li></a> <a ><li class="rz-nav-item"><span>促销活动</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>订单处理</h2> <table> <thead> <tr> <th>订单编号</th> <th>购买商品</th> <th>订单状态</th> <th>支付情况</th> <th>发货信息</th> <th>操作</th> </tr> </thead> <tbody> <tr> <td>123456</td> <td>商品A</td> <td>待发货</td> <td>已支付</td> <td>无</td> <td class="action-buttons"> <button class="refund-button">申请退款</button> <button class="complaint-button">投诉</button> </td> </tr> <tr> <td>123457</td> <td>商品B</td> <td>已发货</td> <td>已支付</td> <td>快递公司：顺丰</td> <td class="action-buttons"> <button class="refund-button">申请退款</button> <button class="complaint-button">投诉</button> </td> </tr> <tr> <td>123458</td> <td>商品C</td> <td>已完成</td> <td>已支付</td> <td>快递公司：圆通</td> <td class="action-buttons"> <button class="refund-button">申请退款</button> <button class="complaint-button">投诉</button> </td> </tr> </tbody> </table> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>高豪商城平台</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>高豪商城平台</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>商品浏览</span></li></a> <a ><li class="rz-nav-item"><span>购物车管理</span></li></a> <a ><li class="rz-nav-item"><span>订单处理</span></li></a> <a ><li class="rz-nav-item active"><span>支付功能</span></li></a> <a ><li class="rz-nav-item"><span>客户评价</span></li></a> <a ><li class="rz-nav-item"><span>促销活动</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>支付功能</h2> <p>本页面提供多种支付方式供用户选择，包括信用卡、支付宝和微信支付等。用户可在确认订单信息后，选择合适的支付方式并填写相关信息。页面确保支付过程安全、快捷，用户完成支付后将收到订单确认信息。</p> <div class="payment-method"> <h3>选择支付方式:</h3> <label> <input type="radio" name="payment" value="credit\_card"> 信用卡 </label> <label> <input type="radio" name="payment" value="alipay"> 支付宝 </label> <label> <input type="radio" name="payment" value="wechat\_pay"> 微信支付 </label> </div> <div class="order-info"> <h3>订单信息:</h3> <p>商品名称: XXX</p> <p>总金额: ¥XXX</p> </div> <div class="payment-details"> <h3>填写支付信息:</h3> <label for="card-number">卡号:</label> <input type="text" id="card-number" placeholder="请输入卡号"> <label for="expiry-date">有效期:</label> <input type="text" id="expiry-date" placeholder="MM/YY"> <label for="cvv">CVV:</label> <input type="text" id="cvv" placeholder="CVV"> </div> <button type="button">确认支付</button> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>高豪商城平台</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>高豪商城平台</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>商品浏览</span></li></a> <a ><li class="rz-nav-item"><span>购物车管理</span></li></a> <a ><li class="rz-nav-item"><span>订单处理</span></li></a> <a ><li class="rz-nav-item"><span>支付功能</span></li></a> <a ><li class="rz-nav-item"><span>客户评价</span></li></a> <a ><li class="rz-nav-item active"><span>促销活动</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>当前促销活动</h2> <div class="promotion-item"> <div class="promotion-title">夏季大促销</div> <div class="promotion-details">全场商品享受8折优惠，活动时间：2023年6月1日至2023年6月30日。</div> <a class="btn">查看详情</a> </div> <div class="promotion-item"> <div class="promotion-title">满减活动</div> <div class="promotion-details">满300元减50元，活动时间：2023年7月1日至2023年7月15日。</div> <a class="btn">查看详情</a> </div> <div class="promotion-item"> <div class="promotion-title">优惠券派发</div> <div class="promotion-details">领取50元优惠券，限新用户使用，活动时间：2023年8月1日至2023年8月31日。</div> <a class="btn">查看详情</a> </div> </div> </main> </div></body></html>

/\*\* \* @description products \* @author 大大的盆 \*/@RestController@RequestMapping(value = "/products")public class ProductsController { @Resource private ProductsService productsService; /\*\* \* 新增 \* @author 大大的盆 \*\*/ @RequestMapping("/insert") public Object insert(Products products){ return productsService.insert(products); } /\*\* \* 刪除 \* @author 大大的盆 \*\*/ @RequestMapping("/delete") public Object delete(int id){ return productsService.delete(id); } /\*\* \* 更新 \* @author 大大的盆 \*\*/ @RequestMapping("/update") public Object update(Products products){ return productsService.update(products); } /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ @RequestMapping("/load") public Object load(int id){ return productsService.load(id); } /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ @RequestMapping("/pageList") public Map<String, Object> pageList(@RequestParam(required = false, defaultValue = "0") int offset, @RequestParam(required = false, defaultValue = "10") int pagesize) { return productsService.pageList(offset, pagesize); }}/\*\* \* @description products \* @author 大大的盆 \*/public interface ProductsService { /\*\* \* 新增 \*/ public Object insert(Products products); /\*\* \* 删除 \*/ public Object delete(int id); /\*\* \* 更新 \*/ public Object update(Products products); /\*\* \* 根据主键 id 查询 \*/ public Products load(int id); /\*\* \* 分页查询 \*/ public Map<String,Object> pageList(int offset, int pagesize);}/\*\* \* @description products \* @author 大大的盆 \*/@Servicepublic class ProductsServiceImpl implements ProductsService { @Resource private ProductsMapper productsMapper; @Override public Object insert(Products products) { // valid if (products == null) { return ReturnT.error("必要参数缺失"); } productsMapper.insert(products); return ReturnT.success(); } @Override public Object delete(int id) { int ret = productsMapper.delete(id); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Object update(Products products) { int ret = productsMapper.update(products); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Products load(int id) { return productsMapper.load(id); } @Override public Map<String,Object> pageList(int offset, int pagesize) { List<Products> pageList = productsMapper.pageList(offset, pagesize); int totalCount = productsMapper.pageListCount(offset, pagesize); // result Map<String, Object> result = new HashMap<String, Object>(); result.put("pageList", pageList); result.put("totalCount", totalCount); return result; }}/\*\* \* @description products \* @author 大大的盆 \*/@Mapper@Repositorypublic interface ProductsMapper { /\*\* \* 新增 \* @author 大大的盆 \*\*/ int insert(Products products); /\*\* \* 刪除 \* @author 大大的盆 \*\*/ int delete(int id); /\*\* \* 更新 \* @author 大大的盆 \*\*/ int update(Products products); /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ Products load(int id); /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ List<Products> pageList(int offset,int pagesize); /\*\* \* 查询 分页查询 count \* @author 大大的盆 \*\*/ int pageListCount(int offset,int pagesize);}<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd"><mapper namespace="HBJZGCXY.com.dao.ProductsMapper"> <resultMap id="BaseResultMap" type="HBJZGCXY.com.entity.Products" > <result column="product\_name" property="productName" /> <result column="product\_description" property="productDescription" /> <result column="price" property="price" /> <result column="stock" property="stock" /> <result column="create\_time" property="createTime" /> <result column="create\_user" property="createUser" /> <result column="update\_time" property="updateTime" /> <result column="update\_user" property="updateUser" /> </resultMap> <sql id="Base\_Column\_List"> product\_name, product\_description, price, stock, create\_time, create\_user, update\_time, update\_user </sql> <insert id="insert" useGeneratedKeys="true" keyColumn="id" keyProperty="id" parameterType="HBJZGCXY.com.entity.Products"> INSERT INTO products <trim prefix="(" suffix=")" suffixOverrides=","> <if test="null != productName and '' != productName"> product\_name, </if> <if test="null != productDescription and '' != productDescription"> product\_description, </if> <if test="null != price and '' != price"> price, </if> <if test="null != stock and '' != stock"> stock, </if> <if test="null != createTime and '' != createTime"> create\_time, </if> <if test="null != createUser and '' != createUser"> create\_user, </if> <if test="null != updateTime and '' != updateTime"> update\_time, </if> <if test="null != updateUser and '' != updateUser"> update\_user </if> </trim> <trim prefix="values (" suffix=")" suffixOverrides=","> <if test="null != productName and '' != productName"> #{productName}, </if> <if test="null != productDescription and '' != productDescription"> #{productDescription}, </if> <if test="null != price and '' != price"> #{price}, </if> <if test="null != stock and '' != stock"> #{stock}, </if> <if test="null != createTime and '' != createTime"> #{createTime}, </if> <if test="null != createUser and '' != createUser"> #{createUser}, </if> <if test="null != updateTime and '' != updateTime"> #{updateTime}, </if> <if test="null != updateUser and '' != updateUser"> #{updateUser} </if> </trim> </insert> <delete id="delete" > DELETE FROM products WHERE id = #{id} </delete> <update id="update" parameterType="HBJZGCXY.com.entity.Products"> UPDATE products <set> <if test="null != productName and '' != productName">product\_name = #{productName},</if> <if test="null != productDescription and '' != productDescription">product\_description = #{productDescription},</if> <if test="null != price and '' != price">price = #{price},</if> <if test="null != stock and '' != stock">stock = #{stock},</if> <if test="null != createTime and '' != createTime">create\_time = #{createTime},</if> <if test="null != createUser and '' != createUser">create\_user = #{createUser},</if> <if test="null != updateTime and '' != updateTime">update\_time = #{updateTime},</if> <if test="null != updateUser and '' != updateUser">update\_user = #{updateUser}</if> </set> WHERE id = #{id} </update> <select id="load" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM products WHERE id = #{id} </select> <select id="pageList" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM products LIMIT #{offset}, #{pageSize} </select> <select id="pageListCount" resultType="java.lang.Integer"> SELECT count(1) FROM products </select></mapper>/\*\* \* @description products \* @author 大大的盆 \*/public class Products implements Serializable { private static final long serialVersionUID = 1L; private String productName; private String productDescription; private String price; private String stock; private String createTime; private String createUser; private String updateTime; private String updateUser; public Products() { } public String getProductName() { return productName; } public void setProductName(String productName) { this.productName = productName; } public String getProductDescription() { return productDescription; } public void setProductDescription(String productDescription) { this.productDescription = productDescription; } public String getPrice() { return price; } public void setPrice(String price) { this.price = price; } public String getStock() { return stock; } public void setStock(String stock) { this.stock = stock; } public String getCreateTime() { return createTime; } public void setCreateTime(String createTime) { this.createTime = createTime; } public String getCreateUser() { return createUser; } public void setCreateUser(String createUser) { this.createUser = createUser; } public String getUpdateTime() { return updateTime; } public void setUpdateTime(String updateTime) { this.updateTime = updateTime; } public String getUpdateUser() { return updateUser; } public void setUpdateUser(String updateUser) { this.updateUser = updateUser; }}<template> <div class="mod-config"> <el-form :inline="true" :model="dataForm" @keyup.enter.native="getDataList()"> <el-form-item> <el-input v-model="dataForm.key" placeholder="参数名" clearable></el-input> </el-form-item> <el-form-item> <el-button @click="getDataList()">查询</el-button> <el-button v-if="isAuth('generator:products:save')" type="primary" @click="addOrUpdateHandle()">新增</el-button> <el-button v-if="isAuth('generator:products:delete')" type="danger" @click="deleteHandle()" :disabled="dataListSelections.length <= 0">批量删除</el-button> </el-form-item> </el-form> <el-table :data="dataList" border v-loading="dataListLoading" @selection-change="selectionChangeHandle" style="width: 100%;"> <el-table-column type="selection" header-align="center" align="center" width="50"> </el-table-column> <el-table-column prop="productName" header-align="center" align="center" label="product\_name"> </el-table-column> <el-table-column prop="productDescription" header-align="center" align="center" label="product\_description"> </el-table-column> <el-table-column prop="price" header-align="center" align="center" label="price"> </el-table-column> <el-table-column prop="stock" header-align="center" align="center" label="stock"> </el-table-column> <el-table-column prop="createTime" header-align="center" align="center" label="create\_time"> </el-table-column> <el-table-column prop="createUser" header-align="center" align="center" label="create\_user"> </el-table-column> <el-table-column prop="updateTime" header-align="center" align="center" label="update\_time"> </el-table-column> <el-table-column prop="updateUser" header-align="center" align="center" label="update\_user"> </el-table-column> <el-table-column fixed="right" header-align="center" align="center" width="150" label="操作"> <template slot-scope="scope"> <el-button type="text" size="small" @click="addOrUpdateHandle(scope.row.productsId)">修改</el-button> <el-button type="text" size="small" @click="deleteHandle(scope.row.productsId)">删除</el-button> </template> </el-table-column> </el-table> <el-pagination @size-change="sizeChangeHandle" @current-change="currentChangeHandle" :current-page="pageIndex" :page-sizes="[10, 20, 50, 100]" :page-size="pageSize" :total="totalPage" layout="total, sizes, prev, pager, next, jumper"> </el-pagination> <!-- 弹窗, 新增 / 修改 --> <add-or-update v-if="addOrUpdateVisible" ref="addOrUpdate" @refreshDataList="getDataList"></add-or-update> </div></template><script> import AddOrUpdate from './products-add-or-update' export default { data () { return { dataForm: { key: '' }, dataList: [], pageIndex: 1, pageSize: 10, totalPage: 0, dataListLoading: false, dataListSelections: [], addOrUpdateVisible: false } }, components: { AddOrUpdate }, activated () { this.getDataList() }, methods: { // 获取数据列表 getDataList () { this.dataListLoading = true this.$http({ url: this.$http.adornUrl('/generator/products/list'), method: 'get', params: this.$http.adornParams({ 'page': this.pageIndex, 'limit': this.pageSize, 'key': this.dataForm.key }) }).then(({data}) => { if (data && data.code === 0) { this.dataList = data.page.list this.totalPage = data.page.totalCount } else { this.dataList = [] this.totalPage = 0 } this.dataListLoading = false }) }, // 每页数 sizeChangeHandle (val) { this.pageSize = val this.pageIndex = 1 this.getDataList() }, // 当前页 currentChangeHandle (val) { this.pageIndex = val this.getDataList() }, // 多选 selectionChangeHandle (val) { this.dataListSelections = val }, // 新增 / 修改 addOrUpdateHandle (id) { this.addOrUpdateVisible = true this.$nextTick(() => { this.$refs.addOrUpdate.init(id) }) }, // 删除 deleteHandle (id) { var ids = id ? [id] : this.dataListSelections.map(item => { return item.productsId }) this.$confirm(`确定对[id=${ids.join(',')}]进行[${id ? '删除' : '批量删除'}]操作?`, '提示', { confirmButtonText: '确定', cancelButtonText: '取消', type: 'warning' }).then(() => { this.$http({ url: this.$http.adornUrl('/generator/products/delete'), method: 'post', data: this.$http.adornData(ids, false) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.getDataList() } }) } else { this.$message.error(data.msg) } }) }) } } }</script><template> <el-dialog :title="!dataForm.productsId ? '新增' : '修改'" :close-on-click-modal="false" :visible.sync="visible"> <el-form :model="dataForm" :rules="dataRule" ref="dataForm" @keyup.enter.native="dataFormSubmit()" label-width="80px"> <el-form-item label="product\_name" prop="productName"> <el-input v-model="dataForm.productName" placeholder="product\_name"></el-input> </el-form-item> <el-form-item label="product\_description" prop="productDescription"> <el-input v-model="dataForm.productDescription" placeholder="product\_description"></el-input> </el-form-item> <el-form-item label="price" prop="price"> <el-input v-model="dataForm.price" placeholder="price"></el-input> </el-form-item> <el-form-item label="stock" prop="stock"> <el-input v-model="dataForm.stock" placeholder="stock"></el-input> </el-form-item> <el-form-item label="create\_time" prop="createTime"> <el-input v-model="dataForm.createTime" placeholder="create\_time"></el-input> </el-form-item> <el-form-item label="create\_user" prop="createUser"> <el-input v-model="dataForm.createUser" placeholder="create\_user"></el-input> </el-form-item> <el-form-item label="update\_time" prop="updateTime"> <el-input v-model="dataForm.updateTime" placeholder="update\_time"></el-input> </el-form-item> <el-form-item label="update\_user" prop="updateUser"> <el-input v-model="dataForm.updateUser" placeholder="update\_user"></el-input> </el-form-item> </el-form> <span slot="footer" class="dialog-footer"> <el-button @click="visible = false">取消</el-button> <el-button type="primary" @click="dataFormSubmit()">确定</el-button> </span> </el-dialog></template><script> export default { data () { return { visible: false, dataForm: { productName: '', productDescription: '', price: '', stock: '', createTime: '', createUser: '', updateTime: '', updateUser: '' }, dataRule: { productName: [{ required: true, message: 'product\_name不能为空', trigger: 'blur' }], productDescription: [{ required: true, message: 'product\_description不能为空', trigger: 'blur' }], price: [{ required: true, message: 'price不能为空', trigger: 'blur' }], stock: [{ required: true, message: 'stock不能为空', trigger: 'blur' }], createTime: [{ required: true, message: 'create\_time不能为空', trigger: 'blur' }], createUser: [{ required: true, message: 'create\_user不能为空', trigger: 'blur' }], updateTime: [{ required: true, message: 'update\_time不能为空', trigger: 'blur' }], updateUser: [{ required: true, message: 'update\_user不能为空', trigger: 'blur' }] } } }, methods: { init (id) { this.dataForm.productsId = id || 0 this.visible = true this.$nextTick(() => { this.$refs['dataForm'].resetFields() if (this.dataForm.productsId) { this.$http({ url: this.$http.adornUrl(`/generator/products/info/${this.dataForm.productsId}`), method: 'get', params: this.$http.adornParams() }).then(({data}) => { if (data && data.code === 0) { this.dataForm.productName = data.products.productName this.dataForm.productDescription = data.products.productDescription this.dataForm.price = data.products.price this.dataForm.stock = data.products.stock this.dataForm.createTime = data.products.createTime this.dataForm.createUser = data.products.createUser this.dataForm.updateTime = data.products.updateTime this.dataForm.updateUser = data.products.updateUser } }) } }) }, // 表单提交 dataFormSubmit () { this.$refs['dataForm'].validate((valid) => { if (valid) { this.$http({ url: this.$http.adornUrl(`/generator/products/${this.dataForm.productsId? 'save' : 'update'}`), method: 'post', data: this.$http.adornData({ 'productName': 'productName' || undefined, 'productDescription': 'productDescription' || undefined, 'price': 'price' || undefined, 'stock': 'stock' || undefined, 'createTime': 'createTime' || undefined, 'createUser': 'createUser' || undefined, 'updateTime': 'updateTime' || undefined, 'updateUser': 'updateUser' || undefined }) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.visible = false this.$emit('refreshDataList') } }) } else { this.$message.error(data.msg) } }) } }) } } }</script>/\*\* \* @description users \* @author 大大的盆 \*/@RestController@RequestMapping(value = "/users")public class UsersController { @Resource private UsersService usersService; /\*\* \* 新增 \* @author 大大的盆 \*\*/ @RequestMapping("/insert") public Object insert(Users users){ return usersService.insert(users); } /\*\* \* 刪除 \* @author 大大的盆 \*\*/ @RequestMapping("/delete") public Object delete(int id){ return usersService.delete(id); } /\*\* \* 更新 \* @author 大大的盆 \*\*/ @RequestMapping("/update") public Object update(Users users){ return usersService.update(users); } /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ @RequestMapping("/load") public Object load(int id){ return usersService.load(id); } /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ @RequestMapping("/pageList") public Map<String, Object> pageList(@RequestParam(required = false, defaultValue = "0") int offset, @RequestParam(required = false, defaultValue = "10") int pagesize) { return usersService.pageList(offset, pagesize); }}/\*\* \* @description users \* @author 大大的盆 \*/public interface UsersService { /\*\* \* 新增 \*/ public Object insert(Users users); /\*\* \* 删除 \*/ public Object delete(int id); /\*\* \* 更新 \*/ public Object update(Users users); /\*\* \* 根据主键 id 查询 \*/ public Users load(int id); /\*\* \* 分页查询 \*/ public Map<String,Object> pageList(int offset, int pagesize);}/\*\* \* @description users \* @author 大大的盆 \*/@Servicepublic class UsersServiceImpl implements UsersService { @Resource private UsersMapper usersMapper; @Override public Object insert(Users users) { // valid if (users == null) { return ReturnT.error("必要参数缺失"); } usersMapper.insert(users); return ReturnT.success(); } @Override public Object delete(int id) { int ret = usersMapper.delete(id); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Object update(Users users) { int ret = usersMapper.update(users); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Users load(int id) { return usersMapper.load(id); } @Override public Map<String,Object> pageList(int offset, int pagesize) { List<Users> pageList = usersMapper.pageList(offset, pagesize); int totalCount = usersMapper.pageListCount(offset, pagesize); // result Map<String, Object> result = new HashMap<String, Object>(); result.put("pageList", pageList); result.put("totalCount", totalCount); return result; }}/\*\* \* @description users \* @author 大大的盆 \*/@Mapper@Repositorypublic interface UsersMapper { /\*\* \* 新增 \* @author 大大的盆 \*\*/ int insert(Users users); /\*\* \* 刪除 \* @author 大大的盆 \*\*/ int delete(int id); /\*\* \* 更新 \* @author 大大的盆 \*\*/ int update(Users users); /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ Users load(int id); /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ List<Users> pageList(int offset,int pagesize); /\*\* \* 查询 分页查询 count \* @author 大大的盆 \*\*/ int pageListCount(int offset,int pagesize);}<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd"><mapper namespace="HBJZGCXY.com.dao.UsersMapper"> <resultMap id="BaseResultMap" type="HBJZGCXY.com.entity.Users" > <result column="username" property="username" /> <result column="password\_hash" property="passwordHash" /> <result column="email" property="email" /> <result column="create\_time" property="createTime" /> <result column="create\_user" property="createUser" /> <result column="update\_time" property="updateTime" /> <result column="update\_user" property="updateUser" /> </resultMap> <sql id="Base\_Column\_List"> username, password\_hash, email, create\_time, create\_user, update\_time, update\_user </sql> <insert id="insert" useGeneratedKeys="true" keyColumn="id" keyProperty="id" parameterType="HBJZGCXY.com.entity.Users"> INSERT INTO users <trim prefix="(" suffix=")" suffixOverrides=","> <if test="null != username and '' != username"> username, </if> <if test="null != passwordHash and '' != passwordHash"> password\_hash, </if> <if test="null != email and '' != email"> email, </if> <if test="null != createTime and '' != createTime"> create\_time, </if> <if test="null != createUser and '' != createUser"> create\_user, </if> <if test="null != updateTime and '' != updateTime"> update\_time, </if> <if test="null != updateUser and '' != updateUser"> update\_user </if> </trim> <trim prefix="values (" suffix=")" suffixOverrides=","> <if test="null != username and '' != username"> #{username}, </if> <if test="null != passwordHash and '' != passwordHash"> #{passwordHash}, </if> <if test="null != email and '' != email"> #{email}, </if> <if test="null != createTime and '' != createTime"> #{createTime}, </if> <if test="null != createUser and '' != createUser"> #{createUser}, </if> <if test="null != updateTime and '' != updateTime"> #{updateTime}, </if> <if test="null != updateUser and '' != updateUser"> #{updateUser} </if> </trim> </insert> <delete id="delete" > DELETE FROM users WHERE id = #{id} </delete> <update id="update" parameterType="HBJZGCXY.com.entity.Users"> UPDATE users <set> <if test="null != username and '' != username">username = #{username},</if> <if test="null != passwordHash and '' != passwordHash">password\_hash = #{passwordHash},</if> <if test="null != email and '' != email">email = #{email},</if> <if test="null != createTime and '' != createTime">create\_time = #{createTime},</if> <if test="null != createUser and '' != createUser">create\_user = #{createUser},</if> <if test="null != updateTime and '' != updateTime">update\_time = #{updateTime},</if> <if test="null != updateUser and '' != updateUser">update\_user = #{updateUser}</if> </set> WHERE id = #{id} </update> <select id="load" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM users WHERE id = #{id} </select> <select id="pageList" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM users LIMIT #{offset}, #{pageSize} </select> <select id="pageListCount" resultType="java.lang.Integer"> SELECT count(1) FROM users </select></mapper>/\*\* \* @description users \* @author 大大的盆 \*/public class Users implements Serializable { private static final long serialVersionUID = 1L; private String username; private String passwordHash; private String email; private String createTime; private String createUser; private String updateTime; private String updateUser; public Users() { } public String getUsername() { return username; } public void setUsername(String username) { this.username = username; } public String getPasswordHash() { return passwordHash; } public void setPasswordHash(String passwordHash) { this.passwordHash = passwordHash; } public String getEmail() { return email; } public void setEmail(String email) { this.email = email; } public String getCreateTime() { return createTime; } public void setCreateTime(String createTime) { this.createTime = createTime; } public String getCreateUser() { return createUser; } public void setCreateUser(String createUser) { this.createUser = createUser; } public String getUpdateTime() { return updateTime; } public void setUpdateTime(String updateTime) { this.updateTime = updateTime; } public String getUpdateUser() { return updateUser; } public void setUpdateUser(String updateUser) { this.updateUser = updateUser; }}<template> <div class="mod-config"> <el-form :inline="true" :model="dataForm" @keyup.enter.native="getDataList()"> <el-form-item> <el-input v-model="dataForm.key" placeholder="参数名" clearable></el-input> </el-form-item> <el-form-item> <el-button @click="getDataList()">查询</el-button> <el-button v-if="isAuth('generator:users:save')" type="primary" @click="addOrUpdateHandle()">新增</el-button> <el-button v-if="isAuth('generator:users:delete')" type="danger" @click="deleteHandle()" :disabled="dataListSelections.length <= 0">批量删除</el-button> </el-form-item> </el-form> <el-table :data="dataList" border v-loading="dataListLoading" @selection-change="selectionChangeHandle" style="width: 100%;"> <el-table-column type="selection" header-align="center" align="center" width="50"> </el-table-column> <el-table-column prop="username" header-align="center" align="center" label="username"> </el-table-column> <el-table-column prop="passwordHash" header-align="center" align="center" label="password\_hash"> </el-table-column> <el-table-column prop="email" header-align="center" align="center" label="email"> </el-table-column> <el-table-column prop="createTime" header-align="center" align="center" label="create\_time"> </el-table-column> <el-table-column prop="createUser" header-align="center" align="center" label="create\_user"> </el-table-column> <el-table-column prop="updateTime" header-align="center" align="center" label="update\_time"> </el-table-column> <el-table-column prop="updateUser" header-align="center" align="center" label="update\_user"> </el-table-column> <el-table-column fixed="right" header-align="center" align="center" width="150" label="操作"> <template slot-scope="scope"> <el-button type="text" size="small" @click="addOrUpdateHandle(scope.row.usersId)">修改</el-button> <el-button type="text" size="small" @click="deleteHandle(scope.row.usersId)">删除</el-button> </template> </el-table-column> </el-table> <el-pagination @size-change="sizeChangeHandle" @current-change="currentChangeHandle" :current-page="pageIndex" :page-sizes="[10, 20, 50, 100]" :page-size="pageSize" :total="totalPage" layout="total, sizes, prev, pager, next, jumper"> </el-pagination> <!-- 弹窗, 新增 / 修改 --> <add-or-update v-if="addOrUpdateVisible" ref="addOrUpdate" @refreshDataList="getDataList"></add-or-update> </div></template><script> import AddOrUpdate from './users-add-or-update' export default { data () { return { dataForm: { key: '' }, dataList: [], pageIndex: 1, pageSize: 10, totalPage: 0, dataListLoading: false, dataListSelections: [], addOrUpdateVisible: false } }, components: { AddOrUpdate }, activated () { this.getDataList() }, methods: { // 获取数据列表 getDataList () { this.dataListLoading = true this.$http({ url: this.$http.adornUrl('/generator/users/list'), method: 'get', params: this.$http.adornParams({ 'page': this.pageIndex, 'limit': this.pageSize, 'key': this.dataForm.key }) }).then(({data}) => { if (data && data.code === 0) { this.dataList = data.page.list this.totalPage = data.page.totalCount } else { this.dataList = [] this.totalPage = 0 } this.dataListLoading = false }) }, // 每页数 sizeChangeHandle (val) { this.pageSize = val this.pageIndex = 1 this.getDataList() }, // 当前页 currentChangeHandle (val) { this.pageIndex = val this.getDataList() }, // 多选 selectionChangeHandle (val) { this.dataListSelections = val }, // 新增 / 修改 addOrUpdateHandle (id) { this.addOrUpdateVisible = true this.$nextTick(() => { this.$refs.addOrUpdate.init(id) }) }

}

} <el-form-item label="create\_time" prop="createTime"> <el-input v-model="dataForm.createTime" placeholder="create\_time"></el-input> </el-form-item> <el-form-item label="create\_user" prop="createUser"> <el-input v-model="dataForm.createUser" placeholder="create\_user"></el-input> </el-form-item> <el-form-item label="update\_time" prop="updateTime"> <el-input v-model="dataForm.updateTime" placeholder="update\_time"></el-input> </el-form-item> <el-form-item label="update\_user" prop="updateUser"> <el-input v-model="dataForm.updateUser" placeholder="update\_user"></el-input> </el-form-item> </el-form> <span slot="footer" class="dialog-footer"> <el-button @click="visible = false">取消</el-button> <el-button type="primary" @click="dataFormSubmit()">确定</el-button> </span> </el-dialog></template><script> export default { data () { return { visible: false, dataForm: { userId: '', totalAmount: '', orderStatus: '', createTime: '', createUser: '', updateTime: '', updateUser: '' }, dataRule: { userId: [{ required: true, message: 'user\_id不能为空', trigger: 'blur' }], totalAmount: [{ required: true, message: 'total\_amount不能为空', trigger: 'blur' }], orderStatus: [{ required: true, message: 'order\_status不能为空', trigger: 'blur' }], createTime: [{ required: true, message: 'create\_time不能为空', trigger: 'blur' }], createUser: [{ required: true, message: 'create\_user不能为空', trigger: 'blur' }], updateTime: [{ required: true, message: 'update\_time不能为空', trigger: 'blur' }], updateUser: [{ required: true, message: 'update\_user不能为空', trigger: 'blur' }] } } }, methods: { init (id) { this.dataForm.ordersId = id || 0 this.visible = true this.$nextTick(() => { this.$refs['dataForm'].resetFields() if (this.dataForm.ordersId) { this.$http({ url: this.$http.adornUrl(`/generator/orders/info/${this.dataForm.ordersId}`), method: 'get', params: this.$http.adornParams() }).then(({data}) => { if (data && data.code === 0) { this.dataForm.userId = data.orders.userId this.dataForm.totalAmount = data.orders.totalAmount this.dataForm.orderStatus = data.orders.orderStatus this.dataForm.createTime = data.orders.createTime this.dataForm.createUser = data.orders.createUser this.dataForm.updateTime = data.orders.updateTime this.dataForm.updateUser = data.orders.updateUser } }) } }) }, // 表单提交 dataFormSubmit () { this.$refs['dataForm'].validate((valid) => { if (valid) { this.$http({ url: this.$http.adornUrl(`/generator/orders/${this.dataForm.ordersId? 'save' : 'update'}`), method: 'post', data: this.$http.adornData({ 'userId': 'userId' || undefined, 'totalAmount': 'totalAmount' || undefined, 'orderStatus': 'orderStatus' || undefined, 'createTime': 'createTime' || undefined, 'createUser': 'createUser' || undefined, 'updateTime': 'updateTime' || undefined, 'updateUser': 'updateUser' || undefined }) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.visible = false this.$emit('refreshDataList') } }) } else { this.$message.error(data.msg) } }) } }) } } }</script>/\*\* \* @description reviews \* @author 大大的盆 \*/@RestController@RequestMapping(value = "/reviews")public class ReviewsController { @Resource private ReviewsService reviewsService; /\*\* \* 新增 \* @author 大大的盆 \*\*/ @RequestMapping("/insert") public Object insert(Reviews reviews){ return reviewsService.insert(reviews); } /\*\* \* 刪除 \* @author 大大的盆 \*\*/ @RequestMapping("/delete") public Object delete(int id){ return reviewsService.delete(id); } /\*\* \* 更新 \* @author 大大的盆 \*\*/ @RequestMapping("/update") public Object update(Reviews reviews){ return reviewsService.update(reviews); } /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ @RequestMapping("/load") public Object load(int id){ return reviewsService.load(id); } /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ @RequestMapping("/pageList") public Map<String, Object> pageList(@RequestParam(required = false, defaultValue = "0") int offset, @RequestParam(required = false, defaultValue = "10") int pagesize) { return reviewsService.pageList(offset, pagesize); }}/\*\* \* @description reviews \* @author 大大的盆 \*/public interface ReviewsService { /\*\* \* 新增 \*/ public Object insert(Reviews reviews); /\*\* \* 删除 \*/ public Object delete(int id); /\*\* \* 更新 \*/ public Object update(Reviews reviews); /\*\* \* 根据主键 id 查询 \*/ public Reviews load(int id); /\*\* \* 分页查询 \*/ public Map<String,Object> pageList(int offset, int pagesize);}/\*\* \* @description reviews \* @author 大大的盆 \*/@Servicepublic class ReviewsServiceImpl implements ReviewsService { @Resource private ReviewsMapper reviewsMapper; @Override public Object insert(Reviews reviews) { // valid if (reviews == null) { return ReturnT.error("必要参数缺失"); } reviewsMapper.insert(reviews); return ReturnT.success(); } @Override public Object delete(int id) { int ret = reviewsMapper.delete(id); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Object update(Reviews reviews) { int ret = reviewsMapper.update(reviews); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Reviews load(int id) { return reviewsMapper.load(id); } @Override public Map<String,Object> pageList(int offset, int pagesize) { List<Reviews> pageList = reviewsMapper.pageList(offset, pagesize); int totalCount = reviewsMapper.pageListCount(offset, pagesize); // result Map<String, Object> result = new HashMap<String, Object>(); result.put("pageList", pageList); result.put("totalCount", totalCount); return result; }}/\*\* \* @description reviews \* @author 大大的盆 \*/@Mapper@Repositorypublic interface ReviewsMapper { /\*\* \* 新增 \* @author 大大的盆 \*\*/ int insert(Reviews reviews); /\*\* \* 刪除 \* @author 大大的盆 \*\*/ int delete(int id); /\*\* \* 更新 \* @author 大大的盆 \*\*/ int update(Reviews reviews); /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ Reviews load(int id); /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ List<Reviews> pageList(int offset,int pagesize); /\*\* \* 查询 分页查询 count \* @author 大大的盆 \*\*/ int pageListCount(int offset,int pagesize);}<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd"><mapper namespace="HBJZGCXY.com.dao.ReviewsMapper"> <resultMap id="BaseResultMap" type="HBJZGCXY.com.entity.Reviews" > <result column="product\_id" property="productId" /> <result column="user\_id" property="userId" /> <result column="rating" property="rating" /> <result column="comment" property="comment" /> <result column="create\_time" property="createTime" /> <result column="create\_user" property="createUser" /> <result column="update\_time" property="updateTime" /> <result column="update\_user" property="updateUser" /> </resultMap> <sql id="Base\_Column\_List"> product\_id, user\_id, rating, comment, create\_time, create\_user, update\_time, update\_user </sql> <insert id="insert" useGeneratedKeys="true" keyColumn="id" keyProperty="id" parameterType="HBJZGCXY.com.entity.Reviews"> INSERT INTO reviews <trim prefix="(" suffix=")" suffixOverrides=","> <if test="null != productId and '' != productId"> product\_id, </if> <if test="null != userId and '' != userId"> user\_id, </if> <if test="null != rating and '' != rating"> rating, </if> <if test="null != comment and '' != comment"> comment, </if> <if test="null != createTime and '' != createTime"> create\_time, </if> <if test="null != createUser and '' != createUser"> create\_user, </if> <if test="null != updateTime and '' != updateTime"> update\_time, </if> <if test="null != updateUser and '' != updateUser"> update\_user </if> </trim> <trim prefix="values (" suffix=")" suffixOverrides=","> <if test="null != productId and '' != productId"> #{productId}, </if> <if test="null != userId and '' != userId"> #{userId}, </if> <if test="null != rating and '' != rating"> #{rating}, </if> <if test="null != comment and '' != comment"> #{comment}, </if> <if test="null != createTime and '' != createTime"> #{createTime}, </if> <if test="null != createUser and '' != createUser"> #{createUser}, </if> <if test="null != updateTime and '' != updateTime"> #{updateTime}, </if> <if test="null != updateUser and '' != updateUser"> #{updateUser} </if> </trim> </insert> <delete id="delete" > DELETE FROM reviews WHERE id = #{id} </delete> <update id="update" parameterType="HBJZGCXY.com.entity.Reviews"> UPDATE reviews <set> <if test="null != productId and '' != productId">product\_id = #{productId},</if> <if test="null != userId and '' != userId">user\_id = #{userId},</if> <if test="null != rating and '' != rating">rating = #{rating},</if> <if test="null != comment and '' != comment">comment = #{comment},</if> <if test="null != createTime and '' != createTime">create\_time = #{createTime},</if> <if test="null != createUser and '' != createUser">create\_user = #{createUser},</if> <if test="null != updateTime and '' != updateTime">update\_time = #{updateTime},</if> <if test="null != updateUser and '' != updateUser">update\_user = #{updateUser}</if> </set> WHERE id = #{id} </update> <select id="load" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM reviews WHERE id = #{id} </select> <select id="pageList" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM reviews LIMIT #{offset}, #{pageSize} </select> <select id="pageListCount" resultType="java.lang.Integer"> SELECT count(1) FROM reviews </select></mapper>/\*\* \* @description reviews \* @author 大大的盆 \*/public class Reviews implements Serializable { private static final long serialVersionUID = 1L; private String productId; private String userId; private String rating; private String comment; private String createTime; private String createUser; private String updateTime; private String updateUser; public Reviews() { } public String getProductId() { return productId; } public void setProductId(String productId) { this.productId = productId; } public String getUserId() { return userId; } public void setUserId(String userId) { this.userId = userId; } public String getRating() { return rating; } public void setRating(String rating) { this.rating = rating; } public String getComment() { return comment; } public void setComment(String comment) { this.comment = comment; } public String getCreateTime() { return createTime; } public void setCreateTime(String createTime) { this.createTime = createTime; } public String getCreateUser() { return createUser; } public void setCreateUser(String createUser) { this.createUser = createUser; } public String getUpdateTime() { return updateTime; } public void setUpdateTime(String updateTime) { this.updateTime = updateTime; } public String getUpdateUser() { return updateUser; } public void setUpdateUser(String updateUser) { this.updateUser = updateUser; }}<template> <div class="mod-config"> <el-form :inline="true" :model="dataForm" @keyup.enter.native="getDataList()"> <el-form-item> <el-input v-model="dataForm.key" placeholder="参数名" clearable></el-input> </el-form-item> <el-form-item> <el-button @click="getDataList()">查询</el-button> <el-button v-if="isAuth('generator:reviews:save')" type="primary" @click="addOrUpdateHandle()">新增</el-button> <el-button v-if="isAuth('generator:reviews:delete')" type="danger" @click="deleteHandle()" :disabled="dataListSelections.length <= 0">批量删除</el-button> </el-form-item> </el-form> <el-table :data="dataList" border v-loading="dataListLoading" @selection-change="selectionChangeHandle" style="width: 100%;"> <el-table-column type="selection" header-align="center" align="center" width="50"> </el-table-column> <el-table-column prop="productId" header-align="center" align="center" label="product\_id"> </el-table-column> <el-table-column prop="userId" header-align="center" align="center" label="user\_id"> </el-table-column> <el-table-column prop="rating" header-align="center" align="center" label="rating"> </el-table-column> <el-table-column prop="comment" header-align="center" align="center" label="comment"> </el-table-column> <el-table-column prop="createTime" header-align="center" align="center" label="create\_time"> </el-table-column> <el-table-column prop="createUser" header-align="center" align="center" label="create\_user"> </el-table-column> <el-table-column prop="updateTime" header-align="center" align="center" label="update\_time"> </el-table-column> <el-table-column prop="updateUser" header-align="center" align="center" label="update\_user"> </el-table-column> <el-table-column fixed="right" header-align="center" align="center" width="150" label="操作"> <template slot-scope="scope"> <el-button type="text" size="small" @click="addOrUpdateHandle(scope.row.reviewsId)">修改</el-button> <el-button type="text" size="small" @click="deleteHandle(scope.row.reviewsId)">删除</el-button> </template> </el-table-column> </el-table> <el-pagination @size-change="sizeChangeHandle" @current-change="currentChangeHandle" :current-page="pageIndex" :page-sizes="[10, 20, 50, 100]" :page-size="pageSize" :total="totalPage" layout="total, sizes, prev, pager, next, jumper"> </el-pagination> <!-- 弹窗, 新增 / 修改 --> <add-or-update v-if="addOrUpdateVisible" ref="addOrUpdate" @refreshDataList="getDataList"></add-or-update> </div></template><script> import AddOrUpdate from './reviews-add-or-update' export default { data () { return { dataForm: { key: '' }, dataList: [], pageIndex: 1, pageSize: 10, totalPage: 0, dataListLoading: false, dataListSelections: [], addOrUpdateVisible: false } }, components: { AddOrUpdate }, activated () { this.getDataList() }, methods: { // 获取数据列表 getDataList () { this.dataListLoading = true this.$http({ url: this.$http.adornUrl('/generator/reviews/list'), method: 'get', params: this.$http.adornParams({ 'page': this.pageIndex, 'limit': this.pageSize, 'key': this.dataForm.key }) }).then(({data}) => { if (data && data.code === 0) { this.dataList = data.page.list this.totalPage = data.page.totalCount } else { this.dataList = [] this.totalPage = 0 } this.dataListLoading = false }) }, // 每页数 sizeChangeHandle (val) { this.pageSize = val this.pageIndex = 1 this.getDataList() }, // 当前页 currentChangeHandle (val) { this.pageIndex = val this.getDataList() }, // 多选 selectionChangeHandle (val) { this.dataListSelections = val }, // 新增 / 修改 addOrUpdateHandle (id) { this.addOrUpdateVisible = true this.$nextTick(() => { this.$refs.addOrUpdate.init(id) }) }, // 删除 deleteHandle (id) { var ids = id ? [id] : this.dataListSelections.map(item => { return item.reviewsId }) this.$confirm(`确定对[id=${ids.join(',')}]进行[${id ? '删除' : '批量删除'}]操作?`, '提示', { confirmButtonText: '确定', cancelButtonText: '取消', type: 'warning' }).then(() => { this.$http({ url: this.$http.adornUrl('/generator/reviews/delete'), method: 'post', data: this.$http.adornData(ids, false) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.getDataList() } }) } else { this.$message.error(data.msg) } }) }) } } }</script><template> <el-dialog :title="!dataForm.reviewsId ? '新增' : '修改'" :close-on-click-modal="false" :visible.sync="visible"> <el-form :model="dataForm" :rules="dataRule" ref="dataForm" @keyup.enter.native="dataFormSubmit()" label-width="80px"> <el-form-item label="product\_id" prop="productId"> <el-input v-model="dataForm.productId" placeholder="product\_id"></el-input> </el-form-item> <el-form-item label="user\_id" prop="userId"> <el-input v-model="dataForm.userId" placeholder="user\_id"></el-input> </el-form-item> <el-form-item label="rating" prop="rating"> <el-input v-model="dataForm.rating" placeholder="rating"></el-input> </el-form-item> <el-form-item label="comment" prop="comment"> <el-input v-model="dataForm.comment" placeholder="comment"></el-input> </el-form-item> <el-form-item label="create\_time" prop="createTime"> <el-input v-model="dataForm.createTime" placeholder="create\_time"></el-input> </el-form-item> <el-form-item label="create\_user" prop="createUser"> <el-input v-model="dataForm.createUser" placeholder="create\_user"></el-input> </el-form-item> <el-form-item label="update\_time" prop="updateTime"> <el-input v-model="dataForm.updateTime" placeholder="update\_time"></el-input> </el-form-item> <el-form-item label="update\_user" prop="updateUser"> <el-input v-model="dataForm.updateUser" placeholder="update\_user"></el-input> </el-form-item> </el-form> <span slot="footer" class="dialog-footer"> <el-button @click="visible = false">取消</el-button> <el-button type="primary" @click="dataFormSubmit()">确定</el-button> </span> </el-dialog></template><script> export default { data () { return { visible: false, dataForm: { productId: '', userId: '', rating: '', comment: '', createTime: '', createUser: '', updateTime: '', updateUser: '' }, dataRule: { productId: [{ required: true, message: 'product\_id不能为空', trigger: 'blur' }], userId: [{ required: true, message: 'user\_id不能为空', trigger: 'blur' }], rating: [{ required: true, message: 'rating不能为空', trigger: 'blur' }], comment: [{ required: true, message: 'comment不能为空', trigger: 'blur' }], createTime: [{ required: true, message: 'create\_time不能为空', trigger: 'blur' }], createUser: [{ required: true, message: 'create\_user不能为空', trigger: 'blur' }], updateTime: [{ required: true, message: 'update\_time不能为空', trigger: 'blur' }], updateUser: [{ required: true, message: 'update\_user不能为空', trigger: 'blur' }] } } }, methods: { init (id) { this.dataForm.reviewsId = id || 0 this.visible = true this.$nextTick(() => { this.$refs['dataForm'].resetFields() if (this.dataForm.reviewsId) { this.$http({ url: this.$http.adornUrl(`/generator/reviews/info/${this.dataForm.reviewsId}`), method: 'get', params: this.$http.adornParams() }).then(({data}) => { if (data && data.code === 0) { this.dataForm.productId = data.reviews.productId this.dataForm.userId = data.reviews.userId this.dataForm.rating = data.reviews.rating this.dataForm.comment = data.reviews.comment this.dataForm.createTime = data.reviews.createTime this.dataForm.createUser = data.reviews.createUser this.dataForm.updateTime = data.reviews.updateTime this.dataForm.updateUser = data.reviews.updateUser } }) } }) }, // 表单提交 dataFormSubmit () { this.$refs['dataForm'].validate((valid) => { if (valid) { this.$http({ url: this.$http.adornUrl(`/generator/reviews/${this.dataForm.reviewsId? 'save' : 'update'}`), method: 'post', data: this.$http.adornData({ 'productId': 'productId' || undefined, 'userId': 'userId' || undefined, 'rating': 'rating' || undefined, 'comment': 'comment' || undefined, 'createTime': 'createTime' || undefined, 'createUser': 'createUser' || undefined, 'updateTime': 'updateTime' || undefined, 'updateUser': 'updateUser' || undefined }) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.visible = false this.$emit('refreshDataList') } }) } else { this.$message.error(data.msg) } }) } }) } } }</script>/\*\* \* @description coupons \* @author 大大的盆 \*/@RestController@RequestMapping(value = "/coupons")public class CouponsController { @Resource private CouponsService couponsService; /\*\* \* 新增 \* @author 大大的盆 \*\*/ @RequestMapping("/insert") public Object insert(Coupons coupons){ return couponsService.insert(coupons); } /\*\* \* 刪除 \* @author 大大的盆 \*\*/ @RequestMapping("/delete") public Object delete(int id){ return couponsService.delete(id); } /\*\* \* 更新 \* @author 大大的盆 \*\*/ @RequestMapping("/update") public Object update(Coupons coupons){ return couponsService.update(coupons); } /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ @RequestMapping("/load") public Object load(int id){ return couponsService.load(id); } /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ @RequestMapping("/pageList") public Map<String, Object> pageList(@RequestParam(required = false, defaultValue = "0") int offset, @RequestParam(required = false, defaultValue = "10") int pagesize) { return couponsService.pageList(offset, pagesize); }}/\*\* \* @description coupons \* @author 大大的盆 \*/public interface CouponsService { /\*\* \* 新增 \*/ public Object insert(Coupons coupons); /\*\* \* 删除 \*/ public Object delete(int id); /\*\* \* 更新 \*/ public Object update(Coupons coupons); /\*\* \* 根据主键 id 查询 \*/ public Coupons load(int id); /\*\* \* 分页查询 \*/ public Map<String,Object> pageList(int offset, int pagesize);}/\*\* \* @description coupons \* @author 大大的盆 \*/@Servicepublic class CouponsServiceImpl implements CouponsService { @Resource private CouponsMapper couponsMapper; @Override public Object insert(Coupons coupons) { // valid if (coupons == null) { return ReturnT.error("必要参数缺失"); } couponsMapper.insert(coupons); return ReturnT.success(); } @Override public Object delete(int id) { int ret = couponsMapper.delete(id); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Object update(Coupons coupons) { int ret = couponsMapper.update(coupons); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Coupons load(int id) { return couponsMapper.load(id); } @Override public Map<String,Object> pageList(int offset, int pagesize) { List<Coupons> pageList = couponsMapper.pageList(offset, pagesize); int totalCount = couponsMapper.pageListCount(offset, pagesize); // result Map<String, Object> result = new HashMap<String, Object>(); result.put("pageList", pageList); result.put("totalCount", totalCount); return result; }}/\*\* \* @description coupons \* @author 大大的盆 \*/@Mapper@Repositorypublic interface CouponsMapper { /\*\* \* 新增 \* @author 大大的盆 \*\*/ int insert(Coupons coupons); /\*\* \* 刪除 \* @author 大大的盆 \*\*/ int delete(int id); /\*\* \* 更新 \* @author 大大的盆 \*\*/ int update(Coupons coupons); /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ Coupons load(int id); /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ List<Coupons> pageList(int offset,int pagesize); /\*\* \* 查询 分页查询 count \* @author 大大的盆 \*\*/ int pageListCount(int offset,int pagesize);}<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd"><mapper namespace="HBJZGCXY.com.dao.CouponsMapper"> <resultMap id="BaseResultMap" type="HBJZGCXY.com.entity.Coupons" > <result column="coupon\_code" property="couponCode" /> <result column="discount\_amount" property="discountAmount" /> <result column="expiration\_date" property="expirationDate" /> <result column="create\_time" property="createTime" /> <result column="create\_user" property="createUser" /> <result column="update\_time" property="updateTime" /> <result column="update\_user" property="updateUser" /> </resultMap> <sql id="Base\_Column\_List"> coupon\_code, discount\_amount, expiration\_date, create\_time, create\_user, update\_time, update\_user </sql> <insert id="insert" useGeneratedKeys="true" keyColumn="id" keyProperty="id" parameterType="HBJZGCXY.com.entity.Coupons"> INSERT INTO coupons <trim prefix="(" suffix=")" suffixOverrides=","> <if test="null != couponCode and '' != couponCode"> coupon\_code, </if> <if test="null != discountAmount and '' != discountAmount"> discount\_amount, </if> <if test="null != expirationDate and '' != expirationDate"> expiration\_date, </if> <if test="null != createTime and '' != createTime"> create\_time, </if> <if test="null != createUser and '' != createUser"> create\_user, </if> <if test="null != updateTime and '' != updateTime"> update\_time, </if> <if test="null != updateUser and '' != updateUser"> update\_user </if> </trim> <trim prefix="values (" suffix=")" suffixOverrides=","> <if test="null != couponCode and '' != couponCode"> #{couponCode}, </if> <if test="null != discountAmount and '' != discountAmount"> #{discountAmount}, </if> <if test="null != expirationDate and '' != expirationDate"> #{expirationDate}, </if> <if test="null != createTime and '' != createTime"> #{createTime}, </if> <if test="null != createUser and '' != createUser"> #{createUser}, </if> <if test="null != updateTime and '' != updateTime"> #{updateTime}, </if> <if test="null != updateUser and '' != updateUser"> #{updateUser} </if> </trim> </insert> <delete id="delete" > DELETE FROM coupons WHERE id = #{id} </delete> <update id="update" parameterType="HBJZGCXY.com.entity.Coupons"> UPDATE coupons <set> <if test="null != couponCode and '' != couponCode">coupon\_code = #{couponCode},</if> <if test="null != discountAmount and '' != discountAmount">discount\_amount = #{discountAmount},</if> <if test="null != expirationDate and '' != expirationDate">expiration\_date = #{expirationDate},</if> <if test="null != createTime and '' != createTime">create\_time = #{createTime},</if> <if test="null != createUser and '' != createUser">create\_user = #{createUser},</if> <if test="null != updateTime and '' != updateTime">update\_time = #{updateTime},</if> <if test="null != updateUser and '' != updateUser">update\_user = #{updateUser}</if> </set> WHERE id = #{id} </update> <select id="load" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM coupons WHERE id = #{id} </select> <select id="pageList" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM coupons LIMIT #{offset}, #{pageSize} </select> <select id="pageListCount" resultType="java.lang.Integer"> SELECT count(1) FROM coupons </select></mapper>/\*\* \* @description coupons \* @author 大大的盆 \*/public class Coupons implements Serializable { private static final long serialVersionUID = 1L; private String couponCode; private String discountAmount; private String expirationDate; private String createTime; private String createUser; private String updateTime; private String updateUser; public Coupons() { } public String getCouponCode() { return couponCode; } public void setCouponCode(String couponCode) { this.couponCode = couponCode; } public String getDiscountAmount() { return discountAmount; } public void setDiscountAmount(String discountAmount) { this.discountAmount = discountAmount; } public String getExpirationDate() { return expirationDate; } public void setExpirationDate(String expirationDate) { this.expirationDate = expirationDate; } public String getCreateTime() { return createTime; } public void setCreateTime(String createTime) { this.createTime = createTime; } public String getCreateUser() { return createUser; } public void setCreateUser(String createUser) { this.createUser = createUser; } public String getUpdateTime() { return updateTime; } public void setUpdateTime(String updateTime) { this.updateTime = updateTime; } public String getUpdateUser() { return updateUser; } public void setUpdateUser(String updateUser) { this.updateUser = updateUser; }}<template> <div class="mod-config"> <el-form :inline="true" :model="dataForm" @keyup.enter.native="getDataList()"> <el-form-item> <el-input v-model="dataForm.key" placeholder="参数名" clearable></el-input> </el-form-item> <el-form-item> <el-button @click="getDataList()">查询</el-button> <el-button v-if="isAuth('generator:coupons:save')" type="primary" @click="addOrUpdateHandle()">新增</el-button> <el-button v-if="isAuth('generator:coupons:delete')" type="danger" @click="deleteHandle()" :disabled="dataListSelections.length <= 0">批量删除</el-button> </el-form-item> </el-form> <el-table :data="dataList" border v-loading="dataListLoading" @selection-change="selectionChangeHandle" style="width: 100%;"> <el-table-column type="selection" header-align="center" align="center" width="50"> </el-table-column> <el-table-column prop="couponCode" header-align="center" align="center" label="coupon\_code"> </el-table-column> <el-table-column prop="discountAmount" header-align="center" align="center" label="discount\_amount"> </el-table-column> <el-table-column prop="expirationDate" header-align="center" align="center" label="expiration\_date"> </el-table-column> <el-table-column prop="createTime" header-align="center" align="center" label="create\_time"> </el-table-column> <el-table-column prop="createUser" header-align="center" align="center" label="create\_user"> </el-table-column> <el-table-column prop="updateTime" header-align="center" align="center" label="update\_time"> </el-table-column> <el-table-column prop="updateUser" header-align="center" align="center" label="update\_user"> </el-table-column> <el-table-column fixed="right" header-align="center" align="center" width="150" label="操作"> <template slot-scope="scope"> <el-button type="text" size="small" @click="addOrUpdateHandle(scope.row.couponsId)">修改</el-button> <el-button type="text" size="small" @click="deleteHandle(scope.row.couponsId)">删除</el-button> </template> </el-table-column> </el-table> <el-pagination @size-change="sizeChangeHandle" @current-change="currentChangeHandle" :current-page="pageIndex" :page-sizes="[10, 20, 50, 100]" :page-size="pageSize" :total="totalPage" layout="total, sizes, prev, pager, next, jumper"> </el-pagination> <!-- 弹窗, 新增 / 修改 --> <add-or-update v-if="addOrUpdateVisible" ref="addOrUpdate" @refreshDataList="getDataList"></add-or-update> </div></template><script> import AddOrUpdate from './coupons-add-or-update' export default { data () { return { dataForm: { key: '' }, dataList: [], pageIndex: 1, pageSize: 10, totalPage: 0, dataListLoading: false, dataListSelections: [], addOrUpdateVisible: false } }, components: { AddOrUpdate }, activated () { this.getDataList() }, methods: { // 获取数据列表 getDataList () { this.dataListLoading = true this.$http({ url: this.$http.adornUrl('/generator/coupons/list'), method: 'get', params: this.$http.adornParams({ 'page': this.pageIndex, 'limit': this.pageSize, 'key': this.dataForm.key }) }).then(({data}) => { if (data && data.code === 0) { this.dataList = data.page.list this.totalPage = data.page.totalCount } else { this.dataList = [] this.totalPage = 0 } this.dataListLoading = false }) }, // 每页数 sizeChangeHandle (val) { this.pageSize = val this.pageIndex = 1 this.getDataList() }, // 当前页 currentChangeHandle (val) { this.pageIndex = val this.getDataList() }, // 多选 selectionChangeHandle (val) { this.dataListSelections = val }, // 新增 / 修改 addOrUpdateHandle (id) { this.addOrUpdateVisible = true this.$nextTick(() => { this.$refs.addOrUpdate.init(id) }) }, // 删除 deleteHandle (id) { var ids = id ? [id] : this.dataListSelections.map(item => { return item.couponsId }) this.$confirm(`确定对[id=${ids.join(',')}]进行[${id ? '删除' : '批量删除'}]操作?`, '提示', { confirmButtonText: '确定', cancelButtonText: '取消', type: 'warning' }).then(() => { this.$http({ url: this.$http.adornUrl('/generator/coupons/delete'), method: 'post', data: this.$http.adornData(ids, false) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.getDataList() } }) } else { this.$message.error(data.msg) } }) }) } } }</script><template> <el-dialog :title="!dataForm.couponsId ? '新增' : '修改'" :close-on-click-modal="false" :visible.sync="visible"> <el-form :model="dataForm" :rules="dataRule" ref="dataForm" @keyup.enter.native="dataFormSubmit()" label-width="80px"> <el-form-item label="coupon\_code" prop="couponCode"> <el-input v-model="dataForm.couponCode" placeholder="coupon\_code"></el-input> </el-form-item> <el-form-item label="discount\_amount" prop="discountAmount"> <el-input v-model="dataForm.discountAmount" placeholder="discount\_amount"></el-input> </el-form-item> <el-form-item label="expiration\_date" prop="expirationDate"> <el-input v-model="dataForm.expirationDate" placeholder="expiration\_date"></el-input> </el-form-item> <el-form-item label="create\_time" prop="createTime"> <el-input v-model="dataForm.createTime" placeholder="create\_time"></el-input> </el-form-item> <el-form-item label="create\_user" prop="createUser"> <el-input v-model="dataForm.createUser" placeholder="create\_user"></el-input> </el-form-item> <el-form-item label="update\_time" prop="updateTime"> <el-input v-model="dataForm.updateTime" placeholder="update\_time"></el-input> </el-form-item> <el-form-item label="update\_user" prop="updateUser"> <el-input v-model="dataForm.updateUser" placeholder="update\_user"></el-input> </el-form-item> </el-form> <span slot="footer" class="dialog-footer"> <el-button @click="visible = false">取消</el-button> <el-button type="primary" @click="dataFormSubmit()">确定</el-button> </span> </el-dialog></template><script> export default { data () { return { visible: false, dataForm: { couponCode: '', discountAmount: '', expirationDate: '', createTime: '', createUser: '', updateTime: '', updateUser: '' }, dataRule: { couponCode: [{ required: true, message: 'coupon\_code不能为空', trigger: 'blur' }], discountAmount: [{ required: true, message: 'discount\_amount不能为空', trigger: 'blur' }], expirationDate: [{ required: true, message: 'expiration\_date不能为空', trigger: 'blur' }], createTime: [{ required: true, message: 'create\_time不能为空', trigger: 'blur' }], createUser: [{ required: true, message: 'create\_user不能为空', trigger: 'blur' }], updateTime: [{ required: true, message: 'update\_time不能为空', trigger: 'blur' }], updateUser: [{ required: true, message: 'update\_user不能为空', trigger: 'blur' }] } } }, methods: { init (id) { this.dataForm.couponsId = id || 0 this.visible = true this.$nextTick(() => { this.$refs['dataForm'].resetFields() if (this.dataForm.couponsId) { this.$http({ url: this.$http.adornUrl(`/generator/coupons/info/${this.dataForm.couponsId}`), method: 'get', params: this.$http.adornParams() }).then(({data}) => { if (data && data.code === 0) { this.dataForm.couponCode = data.coupons.couponCode this.dataForm.discountAmount = data.coupons.discountAmount this.dataForm.expirationDate = data.coupons.expirationDate this.dataForm.createTime = data.coupons.createTime this.dataForm.createUser = data.coupons.createUser this.dataForm.updateTime = data.coupons.updateTime this.dataForm.updateUser = data.coupons.updateUser } }) } }) }, // 表单提交 dataFormSubmit () { this.$refs['dataForm'].validate((valid) => { if (valid) { this.$http({ url: this.$http.adornUrl(`/generator/coupons/${this.dataForm.couponsId? 'save' : 'update'}`), method: 'post', data: this.$http.adornData({ 'couponCode': 'couponCode' || undefined, 'discountAmount': 'discountAmount' || undefined, 'expirationDate': 'expirationDate' || undefined, 'createTime': 'createTime' || undefined, 'createUser': 'createUser' || undefined, 'updateTime': 'updateTime' || undefined, 'updateUser': 'updateUser' || undefined }) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.visible = false this.$emit('refreshDataList') } }) } else { this.$message.error(data.msg) } }) } }) } } }</script>